This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings

Generalized DCell Structure for Load-Balanced Data Center Networks

Markus Kliegl*, Jason Lee†, Jun Li‡, Xinchao Zhang§, Chuanxiong Guo♯, David Rincón¶
*Swarthmore College, †Duke University, ‡Fudan University, §Shanghai Jiao Tong University,
♯Microsoft Research Asia, ¶Universitat Politècnica de Catalunya

***Abstract*—DCell [3] has been proposed as a server centric network structure for data centers. DCell can support millions of servers with high network capacity and provide good fault tolerance by only using commodity mini-switches. However, the traffic in DCell is imbalanced in that links at different layers carry very different number of flows. In this paper, we present a generalized DCell framework so that structures with different connecting rules can be constructed. We show that these structures still preserve the desirable properties of the original DCell structure. Furthermore, we show that the new structures are more symmetric and provide much better load-balancing when using shortest-path routing. We demonstrate the load-balancing property of the new structures by extensive simulations.**

I. Introduction

Data centers are becoming increasingly important and complex. For instance, data centers are critical to the operation of companies such as Microsoft, Yahoo!, and Google, which already run data centers with several hundreds of thousands of servers. Furthermore, data center growth exceeds even Moore's Law [9]. It is clear that the traditional tree structure employed for connecting servers in data centers will no longer be sufficient for future cloud computing and distributed computing applications. There is, therefore, an immediate need to design new network topologies that can meet these rapid expansion requirements.

Current network topologies that have been studied for large data centers include fat-tree [10], BCube [2], and FiConn [5]. These three address different issues: For large data centers, fat-tree requires the use of expensive high-end switches to overcome bottlenecks, and is therefore more useful for smaller data centers. BCube is meant for container-based data center networks, which are of the order of only a few thousand servers. FiConn is designed to utilize currently unused backup ports in already existing data center networks.

Guo et al. [3] have recently proposed a novel network structure called DCell, which addresses the needs of a mega data center. Its desirable properties include

- doubly exponential scaling
- high network capacity
- large bisection width
- small diameter
- fault-tolerance
- requires only commodity network components
- supports an efficient and scalable routing algorithm

One problem that remains in DCell is that the load is not evenly balanced among the links in all-to-all communication. This is true for DCellRouting algorithm, a hierarchical routing algorithm by DCell proposed in [3], as well as shortest path routing. This could be an obstacle to the use of the DCell topology.

In this paper, we address the traffic imbalance by showing that DCell is but one member of a family of graphs satisfying all of the good properties listed above, and there are structures within the family that provide much better load-balancing property than the original DCell structure.

After introducing this family of generalized DCell graphs, we explore the graph properties common to all of them as well as some differences between individual members of the family. We provide better bounds than [3] for the number of servers and the diameter of the DCell structures. In particular, we show numerically that the new DCell members provide much smaller diameter than the original DCell structure and we also explore the symmetries of the graphs.

We show simulation results on the path length distribution and flow distribution for both the DCellRouting and shortest path routing for several realistic parameter values. The most important finding here is that other members of the generalized DCell graph family have significantly better load-balancing properties than the original DCell graph.

The rest of the paper is organized as follows. In Sec. II, we introduce the generalized DCell design. In Sec. III, we present our results on the graph properties of generalized DCells. In Sec. IV, we prove results on path length and flow distribution when using DCellRouting. In Sec. V, we present simulation results for path length and flow distribution using shortest path routing and DCellRouting. In Sec. VI, we conclude the paper and outline our work-in-progress to design a load-balanced routing algorithm.

II. Constructing Generalized DCell

A. Structure of Generalized DCell

The general construction principle of the generalized DCell is the same as that of the original DCell [3]. A DCell_0 consists of n servers connected to a common switch—as an abstract graph, we model this as K_n, the complete graph on n vertices, since switches can be regarded as transparent network devices. From here, we proceed recursively. Denote by t_k the number

978-1-4244-6739-6/10/$26.00 ©2010 IEEE

of servers in a DCell_k. Then, to construct a DCell_k, we take $t_{k-1}+1$ DCell_{k-1}'s and connect them in such a way that

(a) there is exactly one edge between every pair of distinct DCell_{k-1}'s, *and*
(b) we have added exactly one edge to each vertex.

Requirement (a) means that, if we contract each DCell_{k-1} to a single point, then the DCell_k is a complete graph on $t_{k-1}+1$ vertices. This imitation of the complete graph is what we believe gives the DCell structure many of its desirable properties. Requirement (b) is the reason why we must have exactly $t_{k-1}+1$ DCell_{k-1}'s in a DCell_k. It ensures that every server has the same number of links and is the reason why DCell scales doubly exponentially in the number of servers.

This is precisely the point of divergence from the original DCell proposal. There, one specific way of meeting requirements (a) and (b) was proposed, which we name the "α connection rule" later on. But there are many other possibilities. Before we can make this idea more precise, we need to discuss how we label the vertices.

Each server is labeled by a vector id $[a_k, a_{k-1}, ..., a_0]$. Here a_k specifies which DCell_{k-1} the server is in; a_{k-1} specifies which DCell_{k-2} inside that DCell_{k-1} the server is in; and so on. So $0 \le a_0 < n$, and for $i \ge 1$, we have $0 \le a_i < t_{i-1}+1$. We can convert a vector id to a scalar *uid* (unique identifier) as follows:

$$u = a_0 + a_1 t_0 + a_2 t_1 + \cdots + a_k t_{k-1}. \tag{1}$$

Note that we have $0 \le u \le t_k - 1$. Most often, we will label servers just by $[a, b]$ where $a \simeq a_k$ is the number of the DCell_{k-1}, and b is the *uid* corresponding to $[a_{k-1}, ..., a_0]$.

Using these notions, we can define mathematically what a connection rule is. Namely, it is a perfect matching ρ_L of the vertices

$$\{0, ..., t_{L-1}\} \times \{0, ..., t_{L-1} - 1\}$$

that must satisfy the following two properties:

1) ρ_L^2 must be the identity, so that the graph is undirected. (This is also implicit in the term "perfect matching".)
2) For all $a \ne c$, there exist b and d such that $\rho_L([a, b]) = [c, d]$. This ensures that there is a L-level link between each pair of distinct DCell_{L-1}'s.

This encapsulates precisely the requirements (a) and (b) above. We summarize the construction in the following definition.

Definition 1: A generalized DCell with parameters $n \ge 2$, $k \ge 0$, and $R = (\rho_1, ..., \rho_k)$ is constructed as follows:

- A DCell_0 is a complete graph on n vertices.
- From here we proceed recursively until we have constructed a DCell_k: A DCell_L consists of $t_{L-1}+1$ DCell_{L-1}'s, where t_{L-1} is the number of vertices in a DCell_{L-1}. Edges are added according to the connection rule ρ_L.

B. Connection Rules

In this section, we give four examples of connection rules. For $n = 2$, $k = 2$, the graphs are shown in Fig. 1.

α. The connection rule for the original DCell is

$$\alpha_L : [a, b] \leftrightarrow \begin{cases} [b+1, a] & \text{if } a \le b, \\ [b, a-1] & \text{if } a > b. \end{cases} \tag{2}$$

β. A mathematically simple connection rule is

$$\beta_L : [a, b] \leftrightarrow [a+b+1 \pmod{t_{L-1}+1}, t_{L-1}-1-b]. \tag{3}$$

γ. For t_{L-1} even, we can leave b unchanged by the switch, except for a change inside the DCell_0.

$$\gamma_L : [a, b] \leftrightarrow \begin{cases} [a+b \pmod{t_{L-1}+1}, b-1] \\ \qquad \text{if } b \text{ is odd}, \\ [a-(b+1) \pmod{t_{L-1}+1}, b+1] \\ \qquad \text{if } b \text{ is even}. \end{cases} \tag{4}$$

δ. For t_{L-1} even:

$$\delta_L : [a, b] \leftrightarrow \begin{cases} [a+b+1 \pmod{t_{L-1}+1}, b+\frac{t_{L-1}}{2}] \\ \qquad \text{if } b < \frac{t_{L-1}}{2}, \\ [a-b+\frac{t_{L-1}}{2}-1 \pmod{t_{L-1}+1}, b-\frac{t_{L-1}}{2}] \\ \qquad \text{otherwise}. \end{cases} \tag{5}$$

In the rest of this paper, when the specific connection rule used is not important, we will speak of just DCells. If we need to make reference to a specific connection rule, we will speak e.g. of α-DCells, meaning DCells with $R = (\alpha_1, ..., \alpha_k)$. In this context, we should clarify why the requirement that t_{k-1} be even is not a practical problem for the γ and δ connection rules. It turns out that t_k is even for $k \ge 1$. Thus, for even n, there is no problem, while for odd n, only a different rule for the 1-level links is needed. Since almost all real-world switches have an even number of ports, we will restrict ourselves to even n whenever convenient.

III. Graph Properties

In this section, we give expressions and bounds for the number of servers and the diameter. We also investigate the symmetries of the different connection rules. Due to the page limitation, we omit the proofs of the theorems in Sec. III and Sec. IV, which can be found in [6].

A. Number of Servers

No closed-form expression for the exact number of servers t_k in a DCell_k is known. However, it is clear from the DCell construction that t_k satisfies the following recurrence relation:

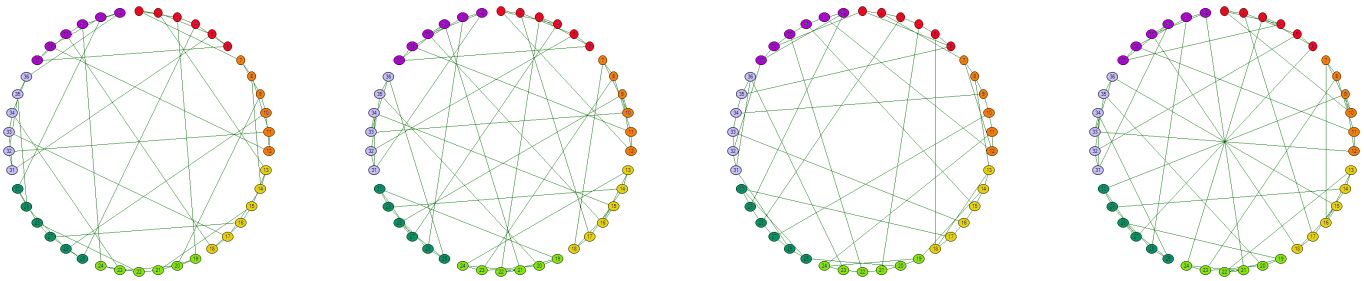
$$\begin{aligned} t_{k+1} &= t_k(t_k+1) \\ t_0 &= n. \end{aligned} \tag{6}$$

This permits t_k to be easily and quickly computed for small n and k. Refer to Table I for values of t_k.

Following a hint by D. E. Knuth in [8], we use the methods of [1] to solve Equation (6), leading to the following theorem.

Theorem 1: We have

$$t_k = \lfloor c^{2^k} \rfloor, \tag{7}$$



(a) α-DCell (b) β-DCell (c) γ-DCell (d) δ-DCell

Fig. 1: Generalized DCells with different connection rules for $n = 2, k = 2$. DCell$_1$ groupings are shown in different colors.

where the constant c is well approximated by the first few terms of the infinite product

$$c = \left(n + \frac{1}{2}\right) \prod_{i=0}^{\infty} \left(1 + \frac{1}{4\left(t_i + \frac{1}{2}\right)^2}\right)^{1/2^{i+1}}. \quad (8)$$

B. Diameter

It is desirable for a data center network to have as small a diameter as possible. For if the diameter is large, then communication between some pairs of servers will necessarily be slow, regardless of the routing algorithm used.

1) An Upper Bound: In [3], it is shown that the diameter of α-DCells satisfies

$$D \leq 2^{k+1} - 1. \quad (9)$$

In fact, the proof carries over easily to all generalized DCells since it uses only the DCellRouting which is independent of the connection rule.

2) A Lower Bound: A well-known (e.g. [4, p.238]) lower bound on the diameter of a graph G with N vertices and maximum degree Δ is

$$D \geq \frac{\log N}{\log \Delta}. \quad (10)$$

Using Theorem 1, this inequality leads to the following theorem.

Theorem 2: The diameter D is bounded below by

$$D \geq 2^k \frac{\log c}{\log(n + k - 1)}. \quad (11)$$

Since c is only slightly larger than $n + \frac{1}{2}$, Theorem 2 can be used to show that

$$D \geq 2^{k-1} \quad (12)$$

when $k \leq n^2 + 1$. Since $n \geq 2$, this includes the realistic cases $k = 3$ and $k = 4$. Together with inequality (9), this narrows the diameter down to within a factor of 4.

3) Dependence on Connection Rule: Table I compares the diameter for different connection rules for some small values of n and k. The diameters of structures constructed by the new connection rules are significantly smaller than that of the original DCell connection rule. For example, for $n = 2$ and $k = 4$, the diameter of the original DCell is 27, whereas it is at most 17 in the new structures. Low diameter leads to higher capacity and also leads to better load-balancing in our case.

n	k	t_k	α	β	γ	δ
2	2	42	7	6	6	6
4	2	440	7	7	7	7
2	3	1,806	15	10	10	10
4	3	176,820	15	13	12	12
2	4	3,263,442	27	17	15	16

TABLE I: Comparison of the diameters of different connection rules.

C. Symmetry

Symmetry is of importance to data center networks primarily because it facilitates the initial wiring.

1) α-DCell: It turns out that, at least for $n \geq 3$, every graph automorphism of a generalized DCell respects its leveled structure; that is, a DCell$_L$ will be mapped to another DCell$_L$ for all L, and all link levels are preserved. Depending on the connection rule, however, there can be much stronger requirements on graph automorphisms. For α-DCell, it appears that there is only one nontrivial symmetry.

Theorem 3: For $k \geq 2$ and $3 \leq n \leq 6$, the automorphism group of an α-DCell$_k$ is isomorphic to C_2, the Cycle group of order 2.

2) Other connection rules: It is straightforward to prove the following theorem.

Theorem 4: Suppose the k-level connection rule of a DCell is of the form:

$$\rho_k : [a, b] \leftrightarrow [a + b + 1 \,(\text{mod } t_{k-1} + 1), g(b)], \quad (13)$$

where g is any permutation on $\{0, ..., t_{k-1} - 1\}$. Then the map

$$\tau : [a, b] \mapsto [a + 1 \,(\text{mod } t_{k-1} + 1), b] \quad (14)$$

is a graph automorphism. τ generates a cyclic subgroup of the automorphism group of order $t_{k-1} + 1$.

Proof: We have

$$\tau([a, b]) = [a + 1 \,(\text{mod } t_{k-1} + 1), b] \quad (15)$$
$$\leftrightarrow [(a + 1) + b + 1 \,(\text{mod } t_{k-1} + 1), g(b)] \quad (16)$$
$$= [(a + b + 1) + 1 \,(\text{mod } t_{k-1} + 1), g(b)] \quad (17)$$
$$= \tau([a + b + 1 \,(\text{mod } t_{k-1} + 1), g(b)]). \quad (18)$$

As for the second assertion, clearly τ is of order $t_{k-1}+1$, since for no smaller number c do we have $a+c \equiv a \pmod{t_{k-1}+1}$. ■

Note that β, γ, and δ are all of this form. Hence, these connection rules lead to significantly more symmetric graphs than the α rule. This group of symmetries consists of exactly the rotational symmetries that are apparent in Fig. 1.

IV. Routing

Since link-state routing is feasible only for small networks [7], it is important to have an efficient, locally computable routing algorithm. In [3], a recursive routing algorithm called DCellRouting is presented for the α-DCell. A similar algorithm can be devised for any generalized DCell. In this section, we state a number of results concerning the path length distribution and flow distribution when using DCellRouting.

A. Path-Length Distribution

As shown in [3], the longest path using DCellRouting is $2^{k+1}-1$.

Fix a vertex v in a DCell$_k$ and let N_i^k denote the number of servers that are exactly i hops away from v in DCellRouting. It turns out that N_i^k is independent of the choice of v, as the following theorem shows.

Theorem 5: N_i^k satisfies

$$N_0^k = 1, \tag{19}$$

$$N_i^0 = \delta_{i0} + (n-1)\delta_{i1}, \tag{20}$$

$$N_i^k = N_i^{k-1} + \sum_{j=0}^{i-1} N_j^{k-1} N_{i-1-j}^{k-1}\ , \qquad \text{for } k, i \geq 1. \tag{21}$$

Here δ_{ij} is the Kronecker delta, which is 1 if $i = j$ and 0 otherwise.

B. Flow Distribution

Theorem 6: In all-to-all communication using DCellRouting, the number of flows F_L carried by a L-level link is

$$F_L = \begin{cases} t_{k-1}^2 & \text{for } L = k, \\ t_{L-1}^2 \prod_{j=L}^{k-1} (1+2t_j) & \text{for } 1 \leq L \leq k-1, \\ (n-1)\prod_{j=0}^{k-1}(1+2t_j) & \text{for } L = 0. \end{cases} \tag{22}$$

Using Theorem 6 and Theorem 1, we can derive from the exact expression for F_L a fairly tight upper bound that is more readily compared to the previously known bound $2^{k-L}t_k$ [3].

Corollary 1: We have

$$F_0 < \frac{n-1}{n+\frac{1}{2}} 2^k (t_k + 0.6) = \frac{n-1}{n+\frac{1}{2}} 2^k t_k (1+o(1)). \tag{23}$$

For $1 \leq L < k$, we have

$$F_L < \frac{t_L - t_{L-1}}{t_L + \frac{1}{2}} 2^{k-L}(t_k + 0.6) = \frac{t_L - t_{L-1}}{t_L + \frac{1}{2}} 2^{k-L} t_k (1+o(1)). \tag{24}$$

n	k	DCR	SP-α	SP-β	SP-γ	SP-δ
2	2	3.73	3.48	3.50	3.46	3.46
4	2	5.16	4.87	4.71	4.68	4.67
6	2	5.73	5.48	5.30	5.26	5.28
8	2	6.04	5.82	5.66	5.59	5.64
2	3	8.18	6.95	6.58	6.44	6.49
4	3	11.29	9.96	8.99	8.68	8.81

(a) Mean

n	k	DCR	SP-α	SP-β	SP-γ	SP-δ
2	2	1.48	1.23	1.25	1.23	1.23
4	2	1.42	1.27	1.15	1.12	1.13
6	2	1.25	1.18	1.09	1.05	1.08
8	2	1.12	1.09	1.04	1.00	1.04
2	3	2.31	1.63	1.41	1.32	1.37
4	3	2.05	1.64	1.22	1.08	1.14

(b) Standard deviation

TABLE II: Expected value and standard deviation of path length distribution. DCR and SP stand for DCellRouting and shortest path routing, respectively.

Finally, we point out that a simple double counting argument shows that the expected value of the path-length distribution is related to the flow distribution as follows.

Theorem 7: The expected value of the path-length distribution is given by

$$E = \frac{\sum_{L=0}^{k} F_L}{t_k - 1}. \tag{25}$$

V. Experimental Results

In this section, we compare empirically the performance of DCellRouting and shortest path routing for the various connection rules. The simulations were necessarily restricted to small n and k; due to the the doubly exponential growth of DCells, these are the only realistic values of n and k.

A. Path-Length Distribution

Table II compares, for some small n and k, the mean and standard deviation of the path length distribution when using DCellRouting or shortest path routing. Shortest path routing for the γ connection rule has the lowest expected value and standard deviation, making it the rule of choice. Fig. 2 shows the different path length distributions for $n = 4$ and $k = 3$. The other cases look similar.

B. Flow Distribution

The flow distributions by link level using shortest path routing and DCellRouting are shown in Fig. 3 for $n = 4, k = 3$. Again, this figure is representative of the other cases as well. We observe that DCellRouting does a poor job of load-balancing. Shortest path routing for α-DCell does better than DCellRouting on average, but has significant bottlenecks that exceed even those of DCellRouting. Shortest path routing for the β, γ, and δ connection rules does better on average and also exhibits very good load-balancing: there are no significant bottleneck links. We believe the asymmetry of α-DCell leads to bottlenecks in the flow distribution and the symmetry in β,δ,γ-DCell leads to

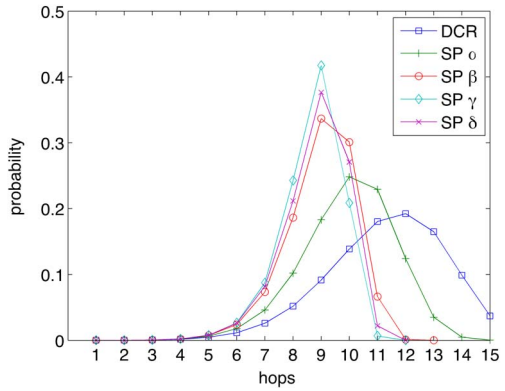
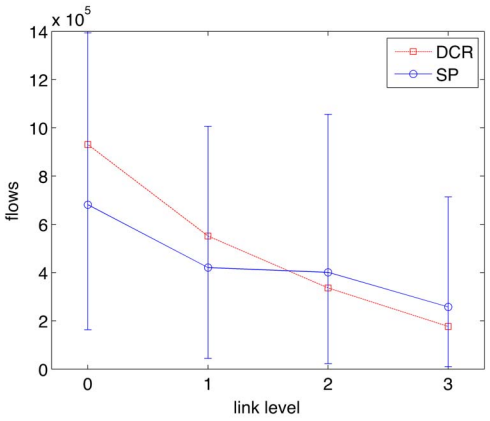
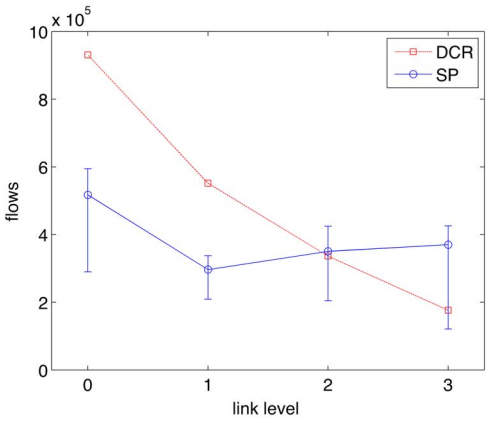


Fig. 2: Path length distribution for $n = 4$ and $k = 3$.



(a) α-DCell



(b) γ-DCell

Fig. 3: Distribution of flows by link level using all-to-all communication for $n = 4$ and $k = 3$. The error bars indicate the maximum and minimum values. The performance of β-DCell and δ-DCell is similar to that of γ-DCell.

balanced flow distribution. It appears that γ is again the rule of choice for all-to-all communication using shortest path routing.

VI. Conclusion and Future Work

In this paper, we present a generalized DCell framework and several structures derived from this framework. We have studied the common graph properties such as scaling, diameter, and symmetry. We also show that the newly introduced structures have much smaller diameter and better load-balancing properties than the original DCell.

The generalized DCell structure introduces a new and huge degree of freedom: the choice of connection rule. We explored only a few of the many possible connection rules—chosen largely for their mathematical simplicity and symmetry—and we are far from having a full understanding of how to design an optimal connection rule. We think the further exploration of this new degree of freedom has the potential to lead to great improvements of the DCell structure. The improved load-balancing and smaller diameters we found are strong evidence for this.

Our future work is to design a practical and scalable shortest-path routing algorithm for the generalized DCell framework. We cannot directly use link-state based routing protocols such as OSPF, since they can only scale to at most a thousand routers. In our future work, we plan to design a shortest-path based routing protocol by taking advantage of the fact that the network topology is known in advance in data centers. We also plan to program servers to handle network failures.

Acknowledgment

This work was performed when Markus Kliegl, Jason Lee, Jun Li, and Xinchao Zhang were visiting students and David Rincón was a visiting academic mentor for the MSRA and UCLA IPAM RIPS-Beijing program at Microsoft Research Asia. The four students are equal contributors. Funding was provided by the NSF and MSRA.

References

[1] A. V. Aho and N. J. A. Sloane. "Some Doubly Exponential Sequences," *Fibonacci Quarterly*, Vol. 11, pp. 429–437, 1970.

[2] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. "BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers," *in Proc. of ACM SIGCOMM*, 2009.

[3] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. "DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers," *in Proc. of ACM SIGCOMM*, 2008.

[4] F. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann, 1992.

[5] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu. "FiConn: Using Backup Port for Server Interconnection in Data Centers," *in Proc. of IEEE INFOCOM*, 2009.

[6] M. Kliegl, J. Lee, J. Li, X. Zhang, C. Guo, D. Rincon. "The Generalized DCell Structures and Their Graph Properties," *Microsoft Research Technical Report,* MSR-TR-2009-140. [Online]. Available: http://research.microsoft.com/apps/pubs/?id=103129

[7] J. T. Moy. *OSPF: Anatomy of an Internet Routing Protocol.* Addison-Wesley Professional, 1998.

[8] N. J. A. Sloane, Ed. Sequence A007018. *The On-Line Encyclopedia of Integer Sequences*, 2009.

[9] J. Snyder. "Microsoft: Datacenter Growth Defies Moore's Law," *PC-World*, 2007. [Online]. Available: http://www.pcworld.com/article/130921/microsoft_datacenter_growth_defies_moores_law.html

[10] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," *in Proc. of ACM SIGCOMM*, 2008.